*Intelligent Information Management*, **2012, 4, 239-250**
http://dx.doi.org/10.4236/iim.2012.425035 Published Online October 2012 (http://www.SciRP.org/journal/iim)

# Finding Optimal Allocation of Constrained Cloud Capacity Using Hyperbolic Voronoi Diagrams on the Sphere

**Shanthi Shanmugam[1], Caroline Shouraboura[2]**
[1]University of California, Berkeley, USA
[2]Forest Ridge School, Seattle, USA
Email: shanthish@forestridge.org, CarolineSh@forestridge.org

Received September 11, 2012; revised October 12, 2012; accepted October 19, 2012

## ABSTRACT

We consider a network of computer data centers on the earth surface delivering computing as a service to a big number of users. The problem is to assign users to data centers to minimize the total communication distance between computing resources and their users in the face of capacity constrained datacenters. In this paper, we extend the classical planar Voronoi Diagram to a hyperbolic Voronoi Diagram on the sphere. We show that a solution to the distance minimization problem under capacity constraints is given by a hyperbolic spherical Voronoi Diagram of data centers. We also present numerical algorithms, computer implementation and results of simulations illustrating our solution. We note applicability of our solution to other important assignment problems, including the assignment of population to regional trauma centers, location of airbases, the distribution of the telecommunication centers for mobile telephones in global telephone companies, and others.

**Keywords:** Cloud Computing; Voronoi Diagram; Voronoi Diagram on the Sphere

## 1. Introduction

Cloud Computing could transform the way we use computer hardware by eliminating the need to own a computer. Instead of placing expensive, fully loaded computers in every school, students all over the world could provision computing resources as they need them. A Gartner research report places the total savings of moving away from individually managed desktop computers at 41%. Behind the Public Cloud is a physical network of data centers with millions of servers, hidden from us through virtualization. Virtualization offers Public Cloud providers the flexibility to assign each application to run on any of its available physical servers.

The vision of less expensive Tablets powered by the remote computing resources of the Cloud is still in its early days. Combined, the emerging Tablet computers coupled with central Cloud Computing facilities to host applications and data will significantly improve schools ability to increase the use of computers in the classroom, allowing students to expand computing power on demand for educational applications anytime anywhere. However, with the transition to the cloud, students' desktops must be moved from local in-classroom computers to large, centralized datacenters. These facilities are typically hundreds, if not thousands, of miles away from the schools and the homes of students. This distance introduces network delay between the students and their desktops. The utility of these students' remote desktops depends on minimizing this network delay.

Presently, the approach cloud providers use to allocate users to a datacenter is referred to as Global Server Load Balancing, or GSLB. A user accesses a computer in the cloud using an Internet domain name. The domain name (for example, www.whitehouse.gov) is translated by a Domain Name Service (DNS) into an Internet Protocol (IP) address pointing to a specific computing device. Where there are multiple datacenters that could satisfy the request, the DNS is programmed to return the IP address of the closest computing device, or the one with the least network delay. The decision of where to put a user is made individually for each connection request. This approach works well for accessing websites where the content is replicated across all cloud locations, such as search engines, social network sites, news, and weather. For student's desktops, a more static assignment is needed. The CAP Theorem [1] states that a distributed system cannot simultaneously provide Consistency, Availability, and Partition Tolerance. Even if students could afford the cost of replicating their desktops continually between the multiple geographically disperse datacenters selected by a latency-based GSLB, the need for desktop consistency over a WAN would exacerbate the impact of Availability events or Partitioning events.

When based in a cloud, students' desktops require a continual static binding to the network access point of the student. The cloud-based desktop should be highly available (for example, using an Availability Zone [2] architecture such as employed by Amazon Web Services). Latency-based GSLBs are not useful for such constant binding of two locations on the earth's surface. A form of GSLB is required which can statically assign students to cloud datacenters.

One other problem remains. Cloud datacenters, despite their large capacity, are finite in size. They can become constrained in capacity. The solution for balancing students across cloud facilities must take into account occasional capacity constraints and allow weighting of assignment to potentially more distant facilities to compensate.

## 2. Problem Formulation

Combined, the emerging tablet computers coupled with central cloud computing facilities to host applications and data significantly improve schools' ability to increase the use of computing in the classroom. Cloud Computing can provide computing as a service to millions of students world-wide. These students would need tablet computers connected to the cloud via common Wi-fi networks and the Internet. The remaining technology required to connect these tablets is a placement service capable of considering proximity and any capacity constraints. Proximity is important due to network delay and the high correlation of this delay to physical distance. Because the relationship between students and their data is continuous, existing GSLB (as described above) will not meet this need. The new placement technology would need to consider occasional capacity constraints as datacenters are physical buildings limited by critical, long-lead-time resources. These include electrical power, cooling capacity and servers, which require many weeks to order, ship, and install. The ideal placement technology could consider an entire population of students and optimally assign it to either the closest datacenter or an alternate should capacity in that datacenter be unavailable. We would like to develop an algorithm which could globally optimize for the variables of population, distance, and specific points of capacity on the earth's surface representing datacenters.

Our starting point in developing a mathematical framework is to model the network of computer datacenters as a set $\mathcal{S}=\{S_1,\cdots,S_n\}$ on the earth's surface, represented by a sphere $\Lambda_R$ of a radius $R>0$. Each datacenter $S_j$ is described by its geographical coordinates, the latitude $\phi_j$ and the longitude $\theta_j$. The population of students (users) is a set $\mathcal{U}=\{U_1,\cdots,U_N\}$. Our task is to create an efficient algorithm to solve the minimization problem for the total distance $\mathcal{D}$ between the data centers and the students. We assume that each student $U_k$ is assigned to a data center $S_{j(k)}$, and the total distance is the sum of the distances between the users and the corresponding data centers,

$$\mathcal{D}(j)=\sum_{k=1}^{N} d\left(S_{j(k)},U_k\right), \tag{1}$$

where $j=j(k)$ is the function assigning the user $U_k$ to the data center $S_{j(k)}$. For any two points *x*, *y* on the sphere we denote $d(x,y)$ the distance between *x* and *y* on the sphere, that is the length of the arc of the great circle connecting *x* to *y*. We also must solve the minimization of the total distance under certain constraints. In real life some data centers become capacity constrained, so the solution must take computing capacity into consideration. We will assume that each data center $S_j$ has a capacity $C_j$ which characterizes how many users it can service. For simplicity we assume here that all users are equal in terms of the requested volume of services. It is easy to extend our model to the case when different users request different volume of services.

The assumption of the limited capacity for the data centers leads to the minimization problem with constraints: find an assignment function $j_0(k)$ such that

$$\min_{j\in\mathcal{J}(C)} \mathcal{D}(j)=\mathcal{D}(j_0), \tag{2}$$

where

$$\mathcal{J}(C)=\left\{j=j(k)\middle| N_m=\#\{k: j(k)=m\}\le C_m \right. \\ \left. \text{for } m=1,\cdots,n\right\}. \tag{3}$$

Observe that $N_m=\#\{k: j(k)=m\}$ is just the number of users assigned to the data center $S_m$. Since the number of users is usually large, it can be useful to consider a (nonnegative) measure $\mu(U)$ of users on the sphere. In this case, the total distance functional looks like

$$\mathcal{D}(j)=\int_{\Lambda_R} d\left(S_{j(U)},U\right)d\mu(U), \tag{4}$$

where $j(U)$ is the assignment function of the users to the data centers.

Respectively, constraint (3) takes the form

$$\mathcal{J}(C)=\left\{j=j(U)\middle| \mu\{U: j(U)=m\}\le C_m \right. \\ \left. \text{for } m=1,\cdots,n\right\}. \tag{5}$$

To solve the minimization problem we explored the possibility of extending classic Voronoi Diagrams. Voronoi Diagrams are a well-known geometric tool for answering distance related queries. Informal use of Voronoi Diagrams can be traced back to Descartes in 1644. In 1854 British physician John Snow used a Voronoi Diagram to illustrate how the majority of people who died in

the Soho cholera epidemic lived closer to the infected Broad Street pump. There have been numerous applications in science and technology since [3-8].

Voronoi Diagrams have the unique property that any point within a Voronoi cell is closer to the vertex of that cell than any other vertex. As we considered how the edges of a Voronoi cell behave in reaction to vertices constrained in the number of points they could service, we found another well-known geometric construct, the hyperbola, to be especially applicable. Our result is described below in Theorem 3.1, where we show that the solution to the minimization problem in the face of capacity constraints is given by a new, extended form of Voronoi Diagram, hyperbolic Voronoi Diagrams on the sphere. We believe this form of Voronoi Diagram could prove useful in solving many problems beyond the student-tablet assignment issue we are tackling here.

## 3. Solution of the Constrained Minimization Problem

Let us begin with the unconstrained minimization problem. In the absence of the capacity constraint, minimization problem (2) can be solved by assigning each user $U$ to its closest data center $S_j$, so that $S_{j_0(U)}$ is the closest data center to $U$. Geometrically, we partition the sphere $\Lambda_R$ into the cells $\{\sigma_m\}$ of the Voronoi Diagram $V$ on the sphere with the points $\{S_m\}$ (**Figure 1**). The cell $\sigma_m$ is defined as the set of points on $\Lambda_R$ which are closer (or at the same distance) to $S_m$ than to other $S_l$'s, that is

$$\sigma_m = \{x \in \Lambda_R : d(x, S_m) \le d(x, S_l) \text{ for all } l \ne m\}. \quad (6)$$

Observe that the cells $\{\sigma_m\}$ are convex spherical polygons on the sphere $\Lambda_R$. With the Voronoi Diagram we associate a graph $\Gamma_V$. The vertices of the graph $\Gamma_V$ are the vertices of the polygons $\{\sigma_m\}$, and the edges of the graph $\Gamma_V$ are the sides of the polygons $\{\sigma_m\}$. The Delaunay Triangulation, associated with the Voronoi Diagram, is the dual graph $\Gamma_D$ to $\Gamma_V$, with vertices $\{S_m\}$ and edges connecting vertices $S_m$, $S_l$ if and only if the cells $\sigma_m$, $\sigma_l$ have a common side.

Thus, in the absence of the constraint, we assign to each data center $S_m$ all users in the cell $\sigma_m$ of the Voronoi Diagram. To describe the minimizing assignment in the presence of the constraint we will introduce hyperbolic Voronoi Diagrams on the sphere. To get an insight into the minimization problem, it's easiest to start by considering the case of a small number of data centers and then formulate a general result. Begin with two data centers.

*Two data centers*. For two data centers $S_1$, $S_2$, the Voronoi Diagram is described by the spherical bisector of $S_1$, $S_2$. The bisector is the arc of the great circle perpendicular to the arc of the great circle through $S_1$, $S_2$ at the middle point (**Figure 2(a)**). The bisector partitions the sphere into two hemispheres, $\sigma_1$, $\sigma_2$, and in the absence of the constraint we assign all users $U_k$ in $\sigma_1$ to $S_1$ and all users in $\sigma_2$ to $S_2$. Let us denote $\mathcal{U}_m$ the set of users in $\sigma_m$, that is

$$\mathcal{U}_m = \{U_k \in \sigma_m\}, m = 1, 2. \quad (7)$$

Suppose now that the data center $S_1$ has a limited capacity $C_1$ and the numbers $N_1$ of the users in $\sigma_1$ is bigger than $C_1$. Some users must be redistributed in the cell $\sigma_1$ from the data center $S_1$ to the one $S_2$ (**Figure 2(b)**). This will increase the total distance by

$$\Delta\mathcal{D}(j) = \sum_{U_k \in \mathcal{U}_{12}} \left[ d(S_2, U_k) - d(S_1, U_k) \right], \quad (8)$$

where $\mathcal{U}_{12} \subset \mathcal{U}_1$ is the set of users redistributed from $S_1$ to $S_2$ (**Figure 3(a)**). The goal is to minimize $\Delta\mathcal{D}(j)$ under the fixed number of the users in $\mathcal{U}_{12}$, since

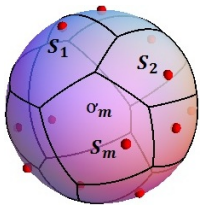



**Figure 1. Voronoi diagram $V$ on the sphere with the points $\{S_m\}$.**

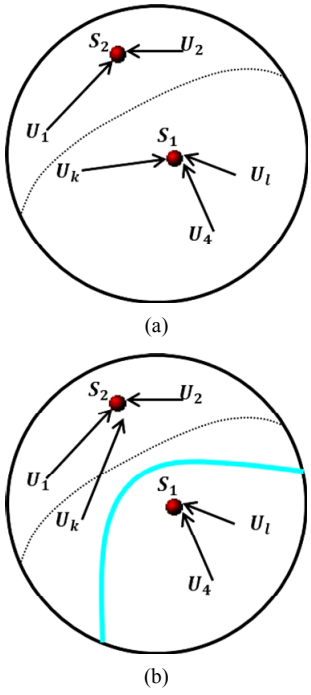



**Figure 2. Two data centers with and without capacity constraints. (a) Without constraints; (b) With constraints.**

$$\left|\mathcal{U}_{12}\right| = N_1 - C_1. \tag{9}$$

This minimization condition implies that if $U_k$, $U_l$ are any two users such that $U_k \in \mathcal{U}_{12}$ and $U_l \in \mathcal{U}_1 \setminus \mathcal{U}_{12}$, then

$$d\left(S_2,U_k\right) - d\left(S_1,U_k\right) \le d\left(S_2,U_l\right) - d\left(S_1,U_l\right), \tag{10}$$

because otherwise we can switch $U_k$ back to $S_1$ and $U_l$ to $S_2$ and in this way decrease $\Delta\mathcal{D}(j)$ in (8). Let

$$\begin{aligned} d_1 &= \max_{U_k \in \mathcal{U}_{12}} \left[ d\left(S_2,U_k\right) - d\left(S_1,U_k\right) \right] \ge 0 \\ &\text{and} \\ d_2 &= \min_{U_k \in \mathcal{U}_1 \setminus \mathcal{U}_{12}} \left[ d\left(S_2,U_k\right) - d\left(S_1,U_k\right) \right]. \end{aligned} \tag{11}$$

Then (10) means that $d_2 \ge d_1$. Let $d$ be any number between $d_1$ and $d_2$. Then

$$\begin{aligned} &\max_{U_k \in \mathcal{U}_{12}} \left[ d\left(S_2,U_k\right) - d\left(S_1,U_k\right) \right] \le d \\ &\le \min_{U_k \in \mathcal{U}_1 \setminus \mathcal{U}_{12}} \left[ d\left(S_2,U_k\right) - d\left(S_1,U_k\right) \right]. \end{aligned} \tag{12}$$

Since $\left[ d\left(S_2,U_k\right) - d\left(S_1,U_k\right) \right] \le 0$ for $U_k \in \mathcal{U}_2$, the latter relation can be extended to

$$\begin{aligned} &\max_{U_k \in \mathcal{U}_{12} \cup \mathcal{U}_2} \left[ d\left(S_2,U_k\right) - d\left(S_1,U_k\right) \right] \le d \\ &\le \min_{U_k \in \mathcal{U}_1 \setminus \mathcal{U}_{12}} \left[ d\left(S_2,U_k\right) - d\left(S_1,U_k\right) \right]. \end{aligned} \tag{13}$$

Consider the spherical hyperbola on the sphere $\Lambda_R$ (cyan line on (**Figure 3(b)**)), defined as a locus of points $x$ satisfying the equation,

$$d\left(x,S_1\right) + d = d\left(x,S_2\right). \tag{14}$$

It partitions the sphere $\Lambda_R$ into two regions $D_1$ and $D_2$ such that $S_1 \in D_1$ and $S_2 \in D_2$ (**Figure 3(b)**). The regions $D_1$ and $D_2$ are described as

$$D_1 = \left\{ x : d\left(x,S_1\right) + d \le d\left(x,S_2\right) \right\} \tag{15}$$

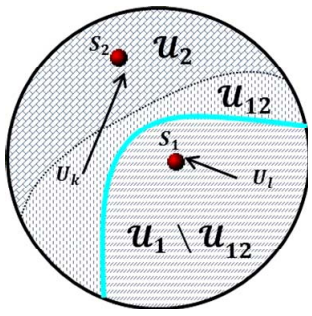
and

$$D_2 = \left\{ x : d\left(x,S_1\right) + d \ge d\left(x,S_2\right) \right\} \tag{16}$$

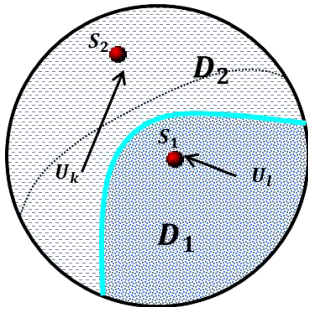
The regions $D_1$, $D_2$ are the cells of the hyperbolic Voronoi Diagram of the points $S_1$, $S_2$ with the parameter $d$. We obtain from (13) that to minimize the total distance $\mathcal{D}(j)$ we have to find such $d$ that the number $N_1$ of users in the region $D_1$ is equal to the capacity $C_1$ and to assign all users in $D_1$ to the data center $S_1$.

Now let's look at the case of three data centers.

*Three data centers*. For three data centers $S_1$, $S_2$, $S_3$ the above considerations prove the following. Suppose that $j_0$ is the minimizing assignment, and let $\mathcal{U}_m$, $m = 1,2,3,$ be the set of users which $j_0$ assigns to $S_m$, $m = 1,2,3$. Then there exists numbers $d_{ml}$ such that for $m \ne l$,




(a)




(b)

**Figure 3. Solution for two data centers with capacity constraints. (a) Illustration for (13); (b) Regions $D_1$, $D_2$.**

$$\begin{aligned} &\max_{U_k \in \mathcal{U}_m} \left[ d\left(S_m,U_k\right) - d\left(S_l,U_k\right) \right] \le d_{ml} \\ &\le \min_{U_k \in \mathcal{U}_l} \left[ d\left(S_m,U_k\right) - d\left(S_l,U_k\right) \right]. \end{aligned} \tag{17}$$

In general, the numbers $d_{ml}$ are not unique, because they can be chosen from the intervals (17). I can state that it is possible to choose these numbers in such a way that

$$d_{12} + d_{23} + d_{31} = 0. \tag{18}$$

Geometrically this equation means that the three spherical hyperbolas, $\gamma_{12}$, $\gamma_{23}$, and $\gamma_{31}$, where

$$\gamma_{ml} = \left\{ x : d\left(S_m,x\right) + d_{ml} = d\left(S_l,x\right) \right\}, \tag{19}$$

intersect in one point. Suppose, for the sake of contradiction, that (18) is not possible. The set of all possible values of the sum $d_{12} + d_{23} + d_{31}$ is the interval,

$$\begin{aligned} &\min d_{12} + \min d_{23} + \min d_{31} \le x \\ &\le \max d_{12} + \max d_{23} + \max d_{31}. \end{aligned} \tag{20}$$

If this interval does not contains 0, then the sum $d_{12} + d_{23} + d_{31}$ is either always positive or always negative. Suppose, for the sake of definiteness, that it is always positive, so that

$$\min d_{12} + \min d_{23} + \min d_{31} = d > 0. \tag{21}$$

Take the minimal values of $d_{12}$, $d_{23}$, $d_{31}$. Then we obtain from (17) that there exists $U_k \in \mathcal{U}_1$, $U_l \in \mathcal{U}_2$, and $U_p \in \mathcal{U}_3$ such that

$$\begin{aligned} d(S_1,U_k)-d(S_2,U_k)&=d_{12},\\ d(S_2,U_l)-d(S_3,U_l)&=d_{23},\\ d(S_3,U_p)-d(S_1,U_p)&=d_{31}. \end{aligned} \tag{22}$$

Let us move $U_k$ from $\mathcal{U}_1$ to $\mathcal{U}_2$, $U_l$ from $\mathcal{U}_2$ to $\mathcal{U}_3$, and $U_p$ from $\mathcal{U}_3$ to $\mathcal{U}_1$. Then the number of users in $\mathcal{U}_1$, $\mathcal{U}_2$, $\mathcal{U}_3$ will not change but the total distance will decrease by

$$\begin{aligned} &\left[d(S_1,U_k)-d(S_2,U_k)\right]+\left[d(S_2,U_l)-d(S_3,U_l)\right]\\ &+\left[d(S_3,U_p)-d(S_1,U_p)\right]=d_{12}+d_{23}+d_{31}=d>0, \end{aligned} \tag{23}$$

hence $j_0$ is not the minimal total distance assignment. This contradiction proves the possibility of choosing the numbers $d_{12}$, $d_{23}$ and $d_{31}$ in such a way that (18) holds.

Equation (18) implies that there exist numbers $d_1$, $d_2$, $d_3$ such that

$$\begin{aligned} d_{12}&=d_1-d_2,\\ d_{23}&=d_2-d_3,\\ d_{31}&=d_3-d_1. \end{aligned} \tag{24}$$

So, by (17),

$$\begin{aligned} \mathcal{U}_m=\{U_k: &d(S_m,U_k)+d_m\le d(S_l,U_k)+d_l\\ &\text{for all } l\ne m\},\ m=1,2,3. \end{aligned} \tag{25}$$

This result can be restated as follows. Partition the sphere $\Lambda_R$ into the three cells,

$$\begin{aligned} D_m=\{x: &d(S_m,x)+d_m\le d(S_l,x)+d_l\\ &\text{for all } l\ne m\},\ m=1,2,3, \end{aligned} \tag{26}$$

of the hyperbolic Voronoi Diagram with the parameters $d_1$, $d_2$, $d_3$. Then to minimize the total distance function assign the users in the cell $D_m$ to the data center $S_m$, $m=1,2,3$.

The parameters $d_1$, $d_2$, $d_3$ are determined by the capacities $C_1$, $C_2$, $C_3$. We assume that the total capacity $C_1+C_2+C_3$ of the data centers exceeds the number of users, because otherwise it would be impossible to service all users. The following three cases can appear:

1) No constraints are active, which happens when the capacities $C_1$, $C_2$, $C_3$ are big enough. In this case $d_1=d_2=d_3=0$, so that $D_m$, $m=1,2,3$, are the classical Voronoi cells on the sphere (**Figure 4(a)**).

2) One constraint is active, say, $C_1$. In this case, $d_1>0$, $d_2=d_3=0$, and the parameter $d_1$ is determined by the condition that the number $N_1$ of users in the cell $D_1$ is equal to $C_1$ (**Figure 4(b)**).

3) Two constraints are active, say, $C_1$, $C_3$. In this case, $d_1>0$, $d_2=0$, $d_3>0$, and the parameters $d_1$, $d_3$ are determined by the conditions that the number $N_m$ of users in the cell $D_m$ is equal to $C_m$ for $m=1,3$ (**Figure 4(c)**).

The subsequent sections present numerical algorithms, computer implementation and results of simulations illustrating the solution for 2, 3, and 4 data centers, and also, for a big number of data centers.

*General case*. The general hyperbolic Voronoi Diagram of points $\{S_1,\cdots,S_n\}$ is defined as follows. Suppose that to each point $S_m$ a number $d_m$ is assigned. Then the *hyperbolic Voronoi Diagram*

$$V(d),\ d=(d_1,\cdots,d_n),$$

with the parameters $\{d_m\}$ and the points $\{S_m\}$, is defined as follows. The cell $D_m$ of the point $S_m$ is defined as

$$\begin{aligned} D_m=\{x\in\Lambda_R: &d(x,S_m)+d_m\le d(x,S_l)+d_l\\ &\text{for all } l\ne m\}. \end{aligned} \tag{27}$$

The curve $\gamma_{ml}$ separating two neighboring cells $D_m$,

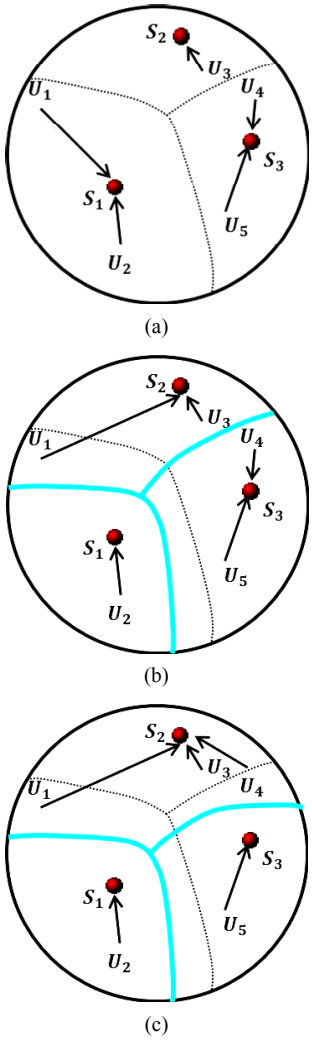



**Figure 4. Three data centers. (a) No constraints; (b) One constraint; (c) Two constraints.**

$D_l$ has the equation,

$$d(x,S_m)+d_m = d(x,S_l)+d_l, \tag{28}$$

so $\gamma_{ml}$ is a part of the spherical hyperbola on the sphere $\Lambda_R$. A vertex $v$ of the hyperbolic Voronoi Diagram is a point which belongs to three or more cells. The graph $\Gamma_V(d)$ of the hyperbolic Voronoi Diagram $V(d)$ consists of the vertices $\{v\}$ and the edges $\{e\}$, which are the curves $\{\gamma_{ml}\}$ separating neighboring cells.

A solution to the constrained minimization problem can be obtained as follows. There is a natural assumption that the total capacity of all data centers is not less than the number of the users,

$$\sum_{m=1}^{n} C_m \geq \mu\Lambda_R. \tag{29}$$

Otherwise, it would be impossible to service all the users.

**Theorem 3.1** *For any measure $\mu(U)$ a minimizer $j_0(U)$ exists, which can be obtained as follows. There exist numbers $(d_1,\cdots,d_n)$ such that the minimizer $j_0$ is obtained by assigning all users in the cell $D_m$ of the hyperbolic Voronoi Diagram $V(d)$ to the data center $S_m$, so that*

$$j_0(U) = m \text{ if and only if } U \in D_m. \tag{30}$$

A full proof of this theorem can be obtained by extending the logic described for three data centers to a general case. We omitted it here due to length.

## 4. Numerical Algorithms and Computer Simulations

For the computer simulations we wrote a system of programs using the MATLAB 2010 software. Specifically, we wrote programs for constructing the Voronoi Diagrams and Delaunay Triangulations on the sphere, both classical and hyperbolic, for calculating the total population in each of the Voronoi cells, and a program with an iterative algorithm for finding a hyperbolic Voronoi Diagram which minimizes the total distance functional under given constraints. We have modeled different types of the constraints:

1) no constraints;
2) limited number of users in the specific cells;
3) equal numbers of users in each Voronoi cells.

In this section we discuss our algorithm in details and present results of our simulations under the constraints (a) and (b) for 2, 3, and 4 data centers. The case of many data centers will be discussed later.

*Two data centers*. $S_1$, $S_2$. As an illustration, start with $S_1$ in Seattle (the magenta diamond in the left upper corner in **Figure 5**) and $S_2$ in Atlanta (the red diamond in **Figure 5**), assuming that all users are located in the USA. Let's also assume that all citizens of the USA are the users. The arc $[S_1,S_2]$ of the great circle connecting $S_1$ to $S_2$ is shown by the yellow line in **Figure 5**. Observe that in **Figure 5** the background is the 3D projection of the earth from Google Earth. The center of the projection is chosen at some point with the latitude $\phi_0$

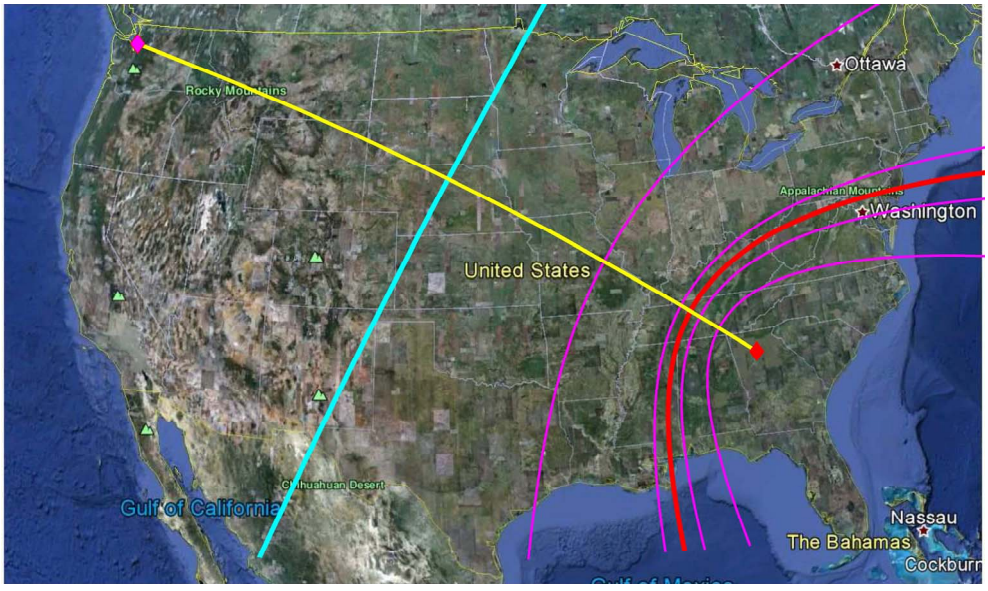



**Figure 5. The solution to the constrained minimization problem with 2 data centers, in Seattle (the magenta diamond in the left upper corner) and Atlanta (the red diamond). The arc of the great circle connecting Seattle to Atlanta is shown by the yellow line. The cyan line is the spherical bisector between Seattle and Atlanta. The thin magenta lines show spherical hyperbolas which are the successive approximations to the hyperbolic Voronoi Diagram on the sphere, with the constraint of 80 million users in the Atlanta data center. The thick red line is the final approximation. The region inside the red line contains 79.4 million citizens, which are assigned to the Atlanta data center.**

and the longitude $\theta_0$. Notice that the great circle connecting Seattle to Atlanta (the yellow line) is not a perfect straight line, and it is slightly curved in the projection.

The spherical bisector $\gamma_0$ of $S_1$, $S_2$,

$$\gamma_0 = \{x \in \Lambda_R : d(x, S_1) = d(x, S_2)\}, \quad (31)$$

(the cyan line in **Figure 5**) is a great circle through the midpoint of the arc $[S_1, S_2]$ and perpendicular to $[S_1, S_2]$. The bisector divides the sphere $\Lambda_R$ into two hemispheres, $\sigma_{\text{Seattle}}$ and $\sigma_{\text{Atlanta}}$. Estimating the population in the two hemispheres suggests that about 73 million citizens live in $\sigma_{\text{Seattle}}$ and about 236 million in $\sigma_{\text{Atlanta}}$. If the goal were to minimize the total distance without any constraint, then the simple answer would be to assign all the users in $\sigma_{\text{Seattle}}$ to the data center in Seattle, and all the users in $\sigma_{\text{Atlanta}}$, to the data center in Atlanta.

But what if the data center in Atlanta has a capacity of only 80 million users? Then we need to redistribute some users from Atlanta to Seattle. Since the goal is to minimize the total distance, the answer is to find a hyperbola

$$\gamma = \{x \in \Lambda_R : d(x, S_1) = d(x, S_2) + d\}, \quad (32)$$

such that the population in $\sigma_{\text{Atlanta}}$ is close to 80 million, but not more. This can be found by successive approximations and by a multiscale analysis of the population distribution.

To analyze the population distribution, start with a large scale distribution, in which the states are represented by their geographic centers and the whole state population is put in the state geographic center. By successive approximations, the hyperbola (32) can be found which gives a large scale solution to the constrained minimization problem. Then, a finer scale is used in which the population of the states near the hyperbola of the large scale solution is represented by smaller units. This results in a solution to the constrained minimization problem at the finer scale. Then, if needed, a second finer scale can be considered, and so on. In the multiscale analysis of the population distribution, we consider big metropolitan areas like New York, Chicago, Houston, etc. as point masses, with all the population of the metropolitan area concentrated in one point.

For the large scale solution to the constrained minimization problem, use successive approximations of the parameter *d* in (32). In the first step $d = d^{(1)}$, where

$$d^{(1)} = \frac{1}{2} d(S_1, S_2) \quad (33)$$

The hyperbola

$$\gamma_1 = \{x \in \Lambda_R : d(x, S_1) = d(x, S_2) + d^{(1)}\} \quad (34)$$

crosses the arc $[S_1, S_2]$ at the point $X_1$ such that

$$d(X_1, S_2) = \frac{1}{4} d(S_1, S_2), \quad (35)$$

see a thin magenta line in **Figure 5**. An analysis of the population distribution gives that about 127 million citizens live in the region $\sigma^{(1)}_{\text{Seattle}}$ and about 182 million in the one $\sigma^{(1)}_{\text{Atlanta}}$, with respect to the hyperbola $\gamma_1$. It is still more than 80 million. Therefore, the second step is to take the hyperbola $\gamma_2$ through the point $X_2$ on the arc $[S_1, S_2]$ such that

$$d(X_2, S_2) = \frac{1}{8} d(S_1, S_2), \quad (36)$$

the second magenta line in **Figure 5**, resulting in about 209.6 million citizens living in the hemisphere $\sigma^{(2)}_{\text{Seattle}}$ and about 99.4 million in the hemisphere $\sigma^{(2)}_{\text{Atlanta}}$, with respect to the hyperbola $\gamma_2$. We continue with this, taking the third, fourth, and subsequent approximations. The first four approximations are shown by thin magenta lines in **Figure 5**. Observe that the successive hyperbolas $\gamma_j$ are jumping back and forth.

We find that the 8-th approximation gives 75.1 million and the 9th one, 83.8 million. After that, all the subsequent approximations oscillate between these two numbers. The reason is the large scale representation of the population. The difference of 8.7 million comes from the state of New Jersey. Therefore, we consider a finer scale for the representation of the population distribution in New Jersey and other states near the hyperbola. In the finer scale we find, by successive approximations, a solution with 79.2 million population inside hyperbola (32). This solution is shown by a thick red line in **Figure 5**. In a similar way, if needed, subsequent approximate solutions with even better approximation to 80 million can be obtained.

*Three data centers*. The next step is to move on to three data centers $S_1$, $S_2$, and $S_3$. As an illustration, we will consider $S_1$ in Seattle, $S_2$ in Atlanta, and $S_3$ in New York, see **Figure 6**.

Focus on the three bisectors,

$$\gamma_{ij} = \{x \in \Lambda_R : d(x, S_i) = d(x, S_j)\}, i \neq j, \quad (37)$$

(the cyan lines in **Figure 6**), which intersect at some point $X_0$ (the red point in **Figure 6**). The point $X_0$ is the vertex of the spherical Voronoi Diagram with the three points $S_1$, $S_2$, and $S_3$. The bisectors $\gamma_{ij}$, emanating from $X_0$ to the opposite point $\pi(X_0)$ on the sphere $\Lambda_R$, are the edges of the Voronoi Diagram. By an analysis of the population distribution, there are about 72.9 million people who live in the cell $\sigma_{\text{Seattle}}$, 144.1 million in $\sigma_{\text{Atlanta}}$, and 92.1 million in $\sigma_{\text{NY}}$. To restrict the population in the Atlanta cell by 80 million, there must be a number *d* such that considering the cells $\sigma_1$, $\sigma_2$, $\sigma_3$ of the hyperbolic Voronoi Diagram,

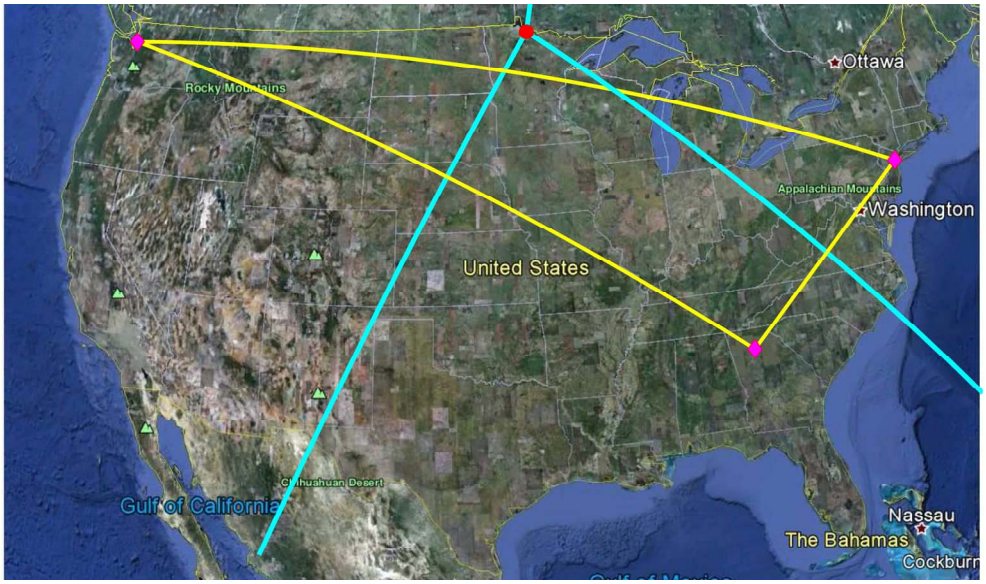



**Figure 6. The spherical Voronoi Diagram for the 3 data centers, in Seattle, Atlanta, and New York. The arcs of the great circles connecting Seattle to Atlanta, Seattle to New York, and Atlanta to New York are shown by yellow lines. The cyan lines are the spherical bisectors between these cities. The cyan lines intersect at a point which is the vertex of the Voronoi Diagram.**

$$\gamma_{ij}=\left\{x\in\Lambda_R : d\left(x,S_i\right)+d_i=d\left(x,S_j\right)+d_j\right\}, i\neq j, \quad (38)$$

with the parameters $\left(d_1,d_2,d_3\right)=\left(0,d,0\right)$, the populations in the Atlanta cell is close to 80 million but under 80 million. This is done by iterations (successive approximations) and by a multiscale representation of the population distribution.

Start with the large scale representation of the population distribution. At the first step of the successive approximations take $d=d^{(1)}$, where

$$d^{(1)}=\frac{1}{2}d\left(S_2,S_3\right) \quad (39)$$

and $d\left(S_2,S_3\right)$ is the distance between Atlanta and New York. This results in a population of 75.4 million in the Seattle cell, 92.4 million in the Atlanta cell, and 141.1 million in the New York cell. Since the population in the Atlanta cell is bigger than 80 million, the second step takes $d=d^{(2)}$, where

$$d^{(2)}=\frac{3}{4}d\left(S_2,S_3\right), \quad (40)$$

and we find the population of 77.4 million in the Seattle cell, 58.3 million in the Atlanta cell, and 173.2 million in the New York cell. The third step takes $d=d^{(3)}$, where

$$d^{(3)}=\frac{5}{8}d\left(S_2,S_3\right), \quad (41)$$

and results in the population of 77.4 million in the Seattle cell, 77.0 million in the Atlanta cell, and 154.5 million in the New York cell. And so on. After several iterations the numbers begin oscillating between two values, and at this point, steps switch to a finer representation of the population distribution near the hyperbolas and continue the iterations.

The final approximation is shown by a thick red line on **Figure 7**. It gives the population of 77.5 million in the Seattle cell, 79.8 million in the Atlanta cell, and 151.7 million in the New York cell.

*Four data centers.* Let us now consider four data centers $S_1$, $S_2$, $S_3$, and $S_4$. As an illustration, assume that $S_1$ is in Seattle, $S_2$ in Atlanta, $S_3$ in New York, and $S_4$ in Phoenix.

**Figure 8** depicts the Delaunay Triangulation on the sphere for $S_1,S_2,S_3,S_4$ (the yellow lines) and edges of the Voronoi Diagram (the cyan lines). The edges of the Voronoi Diagram are the arcs of the bisectors between $S_i$, $S_j$, $i\neq j$. There are 4 Voronoi cells $\left\{\sigma_j\right\}$, so that $S_j\in\sigma_j$. An analysis of the population distribution gives that 30.4 million are in the Seattle cell $\sigma_1$, 141.2 million in the Atlanta cell $\sigma_2$, 92.1 million in the New York cell $\sigma_3$, and 45.3 million in the Phoenix cell $\sigma_4$.

Suppose that the Atlanta data center has a capacity of 80 million, and each of the other centers has a big capacity. Then a new number *d* is required such that considering the cells $\sigma_1$, $\sigma_2$, $\sigma_3$, $\sigma_4$ of the hyperbolic Voronoi Diagram,

$$\gamma_{ij}=\left\{x\in\Lambda_R : d\left(x,S_i\right)+d_i=d\left(x,S_j\right)+d_j\right\}, i\neq j, \quad (42)$$

with the parameters $\left(d_1,d_2,d_3,d_4\right)=\left(0,d,0,0\right)$, the population in the Atlanta cell $\sigma_2$ is close to 80 million but not more. This is done by iterations and by the multiscale representation of the population distribution.

After several successive approximations of the parameter *d*, the population in the Atlanta cell begins to

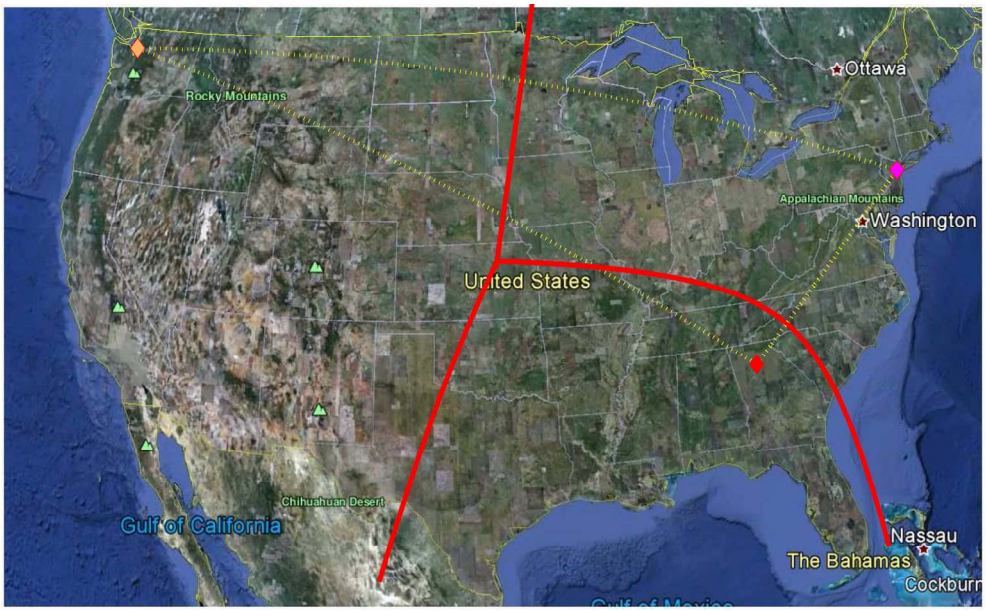



**Figure 7. The solution to the constrained minimization problem with the 3 data centers, in Seattle, Atlanta, and New York. The solution is shown by a thick red line.**

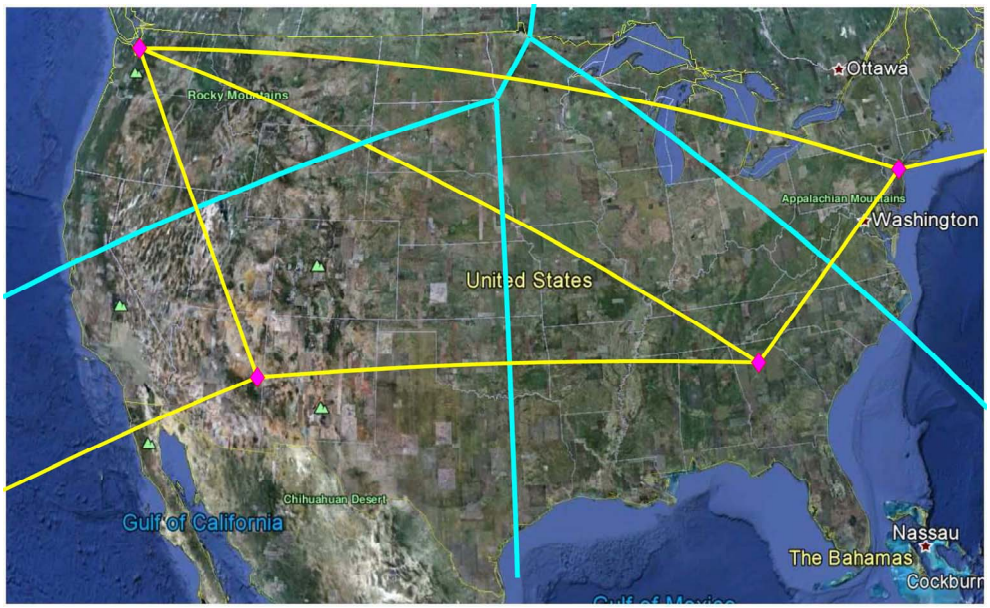



**Figure 8. The Voronoi Diagram on the sphere for the 4 data centers, in Seattle, Atlanta, New York, and Phoenix. The arcs of the Delaunay Triangulation on the sphere for the 4 centers are shown by yellow lines. The cyan lines are the edges of the Voronoi Diagram.**

oscillate between 73.0 million and 86.0 million. An analysis of the population distribution shows that the oscillations are due to the state of Illinois, with the population of 13 million. Therefore, the next step is to consider a finer representation of the population distribution in Illinois and some other states near the edges of the Voronoi Diagram, continuing the iterations of the parameter *d*. Finally, the result is the hyperbolic Voronoi Diagram shown in **Figure 9**, in which the population in the Atlanta cell is 79.3 million, which is close to 80 million. The populations in the other cells are: 30.4 million in the Seattle cell, 130.5 million in the New York cell, and 68.8 million in the Phoenix cell.

It is worth noticing the bifurcation of the hyperbolic Voronoi Diagram during the iterations. Namely, in the original Voronoi Diagram on **Figure 8** the Seattle and Atlanta cells have a common boundary within the figure, while in the final hyperbolic Voronoi Diagram on **Figure**

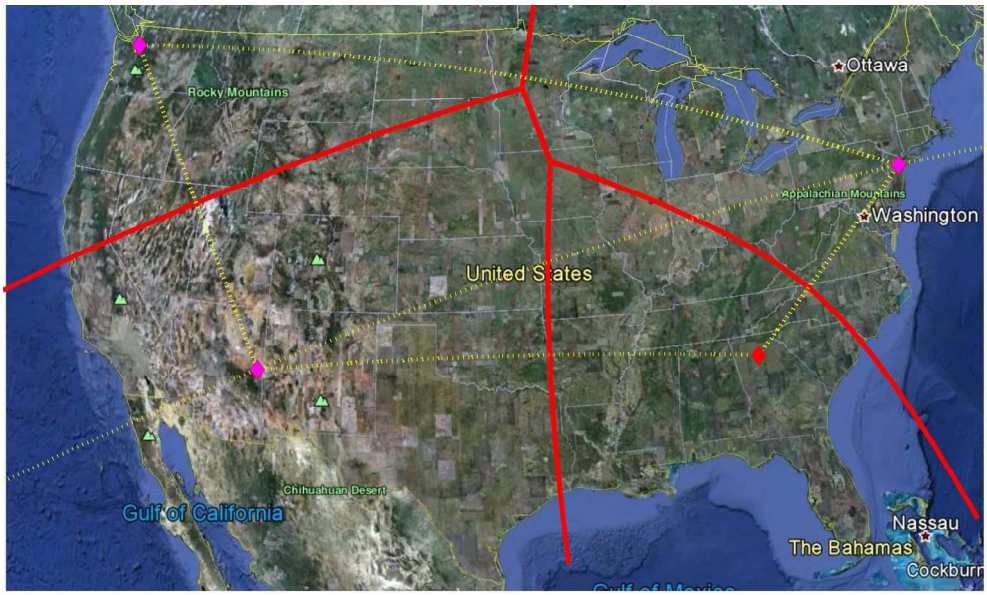



**Figure 9. The solution to the constrained minimization problem with the 4 data centers, in Seattle, Atlanta, New York, and Phoenix. The solution is shown by a thick red line. Observe the bifurcation of the Voronoi Diagram from Figure 8.**

**9** they do not have a common boundary within the figure.

## 5. The Voronoi Diagram on the Sphere and the Convex Hull

Given $n$ points $\{S_1,\cdots,S_n\}$ on the sphere $\Lambda_R$, the Voronoi cell $\sigma_j$ of the point $S_j$ is defined as the set of points which are closer to $S_j$ (or at the same distance) than to any other point $S_i$ with respect to the spherical metric on $\Lambda_R$. The Voronoi cell is a convex spherical polygon and the set of vertices of all Voronoi cells is the set of vertices of the spherical Voronoi Diagram.

There is a nice relation between the spherical Voronoi Diagram and the convex hull $H$ of the points $\{S_1,\cdots,S_n\}$ in the 3D space. This relation was observed by Brown [9], [10] (see also [11]). Namely, looking at any facet $F_m$ of the convex hull $H$ and the plane $P_m$ through $F_m$, the plane $P_m$ intersects the sphere $\Lambda_R$ by a circle $\xi_m$. Then the center $v_m$ of the circle $\xi_m$ on the sphere $\Lambda_R$ is a vertex of the Voronoi Diagram. More precisely, $v_m \in \Lambda_R$ is the center of the spherical cap $\mathcal{C}_m$, bounded by the circle $\xi_m$, which contains no points $S_j$ inside the cap. The set $\{v_m\}$ coincides with the set of vertices of the Voronoi Diagram. This gives a powerful method of the construction of the spherical Voronoi Diagram. In particular, since the convex hull of the points $\{S_1,\cdots,S_n\}$ can be constructed in $\mathcal{O}(n\ln n)$ operations, the spherical Voronoi Diagram can be constructed in $\mathcal{O}(n\ln n)$ operations as well.

Two neighboring facets $F_m$ and $F_l$ of the convex hull share an edge $e_{ml}$ and two vertices $S_i$ and $S_j$, the end-points of the edge $e_{ml}$. The arc of the Delaunay Triangulation connecting $S_i$ and $S_j$ is the arc of the great circle which lies in the dihedral domain formed by the half-planes through $F_m$ and $F_l$, which is vertically opposite to the one containing the convex hull $H$.

## 6. Numerical Algorithm for Many Centers

In the case of many data centers $\{S_1,\cdots,S_n\}$, the first step is construction of their spherical Voronoi Diagram. The next step is to compare the capacity $C_m$ of the data center $S_m$ with the number of users $N_m$ in the Voronoi cell $\sigma_m$ containing $S_m$. The following assumption is made, which seems plausible for the computer cloud networks: for most $S_m$'s the capacity $C_m$ is big enough so it does not create any constraint. Mathematically, there is an assumption that for these $S_m$'s we have an unlimited capacity, $C_m = \infty$. It is also plausible to assume that the cells with constraints are isolated. Both assumptions stem from the practical observation that capacity constraints result in suboptimal operational characteristics for computer clouds, and are therefore uncommon. In this case we need to change the Voronoi Diagram only locally. This can be done similarly to the described above procedure for a small number of data centers.

As an example, look at a model minimization problem for 10 data centers in Seattle, Atlanta, New York, Phoenix, San Francisco, Denver, Houston, Chicago, Boston, and Miami (see **Figure 10**). The Voronoi Diagram on the sphere for these centers is shown by the cyan line. The distribution of the population in the Voronoi cells looks as follows: Seattle—13.8 million, Atlanta—48.5 million, New York—61.5 million, Phoenix—6.7 million, San Francisco—41.1 million, Denver—18.6 million, Houston

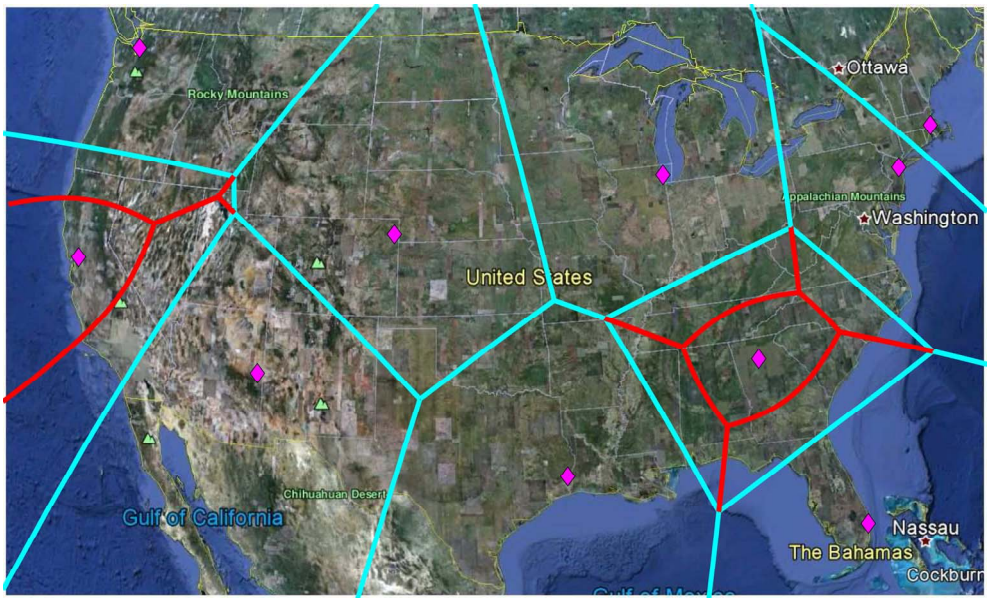



**Figure 10. The solution to the constrained minimization problem with 10 data centers, Seattle, Atlanta, New York, Phoenix, San Francisco, Denver, Houston, Chicago, Boston, and Miami. The Voronoi Diagram on the sphere with these 10 points is shown by the cyan line. In this case, there are two constraints: in San Francisco the capacity is equal to 15 million users and in Atlanta it is 20 million. The constraints change the Voronoi Diagram locally near Atlanta and San Francisco to the hyperbolic diagram. The changes are shown in red.**

—31.5 million, Chicago—57.6 million, Boston—10.9 million, and Miami—18.7 million. Suppose that the data centers in Atlanta and San Francisco have limited capacities of 20 and 15 million users, respectively. Then their users must be redistributed to the neighboring centers. To solve the minimization problem with these constraints, the hyperbolic Voronoi Diagram is first constructed. Then the iterative method described above is applied together with the multiscale analysis of the population. As a result, the hyperbolic Voronoi Diagram on the sphere depicted in **Figure 10** is arrived at. It differs from the initial Voronoi Diagram only locally, near Atlanta and San Francisco. The change from the initial Voronoi Diagram to the new one is shown in red in **Figure 10**.

## 7. Conclusions

In this work we studied the minimization problem for the total communication distance in a computer cloud under the condition of restricted capacity of the data centers. We assumed that the earth is a perfect sphere of radius *R*, so that the earth distance between two points is the length of the smaller arc of the great circle connecting these points. Our main result is Theorem 3.1, which shows that a solution to the minimization problem is given by a hyperbolic Voronoi Diagram constructed on the data centers $S_1,\cdots,S_n$. The parameters $d_1,\cdots,d_n$ of the hyperbolic Voronoi Diagram can be found from the condition that the number of users in each cell $D_j$ of the diagram does not exceed the capacity of the corresponding data center $S_j$.

We discuss numerical algorithms and computer implementation for the construction of the hyperbolic Voronoi Diagrams satisfying the capacity conditions. We consider the numerical solution for a small number of data centers, 2, 3, and 4, and for a large number of data centers. In the latter case we make a plausible assumption that most of the data centers have sufficient capacity to service the clients, and the data centers of insufficient capacity are isolated. This allows local construction of the hyperbolic Voronoi Diagram from a standard Voronoi Diagram.

Although we discuss the application to the computer cloud only, it is interesting to note that our solution and numerical algorithm can be used to solve other important assignment problems. We can mention the problem of location of air-bases [8], the assignment of population to regional trauma centers, the distribution of facilities in global Internet companies like Amazon.com, the distribution of the telecommunication centers for mobile telephones in global telephone companies, data collection centers, and others.

## REFERENCES

[1] S. Gilbert and N. Lynch, "Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services," *ACM SIGACT News*, Vol. 33, No. 2, 2002, pp. 51-59. doi:10.1145/564585.564601

[2] http://docs.amazonwebservices.com/AWSEC2/latest/

UserGuide/using-regions-availability-zones.html

[3] F. Aurenhammer, "Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys*, Vol. 23, No. 3, 1991, pp. 345-405. doi:10.1145/116873.116880

[4] P. Bleher and C. Shouraboura, "Placement of Applications in Computing Clouds Using Voronoi Diagrams," *Journal of Internet Services and Applications*, Vol. 2, No. 3, 2011, pp. 229-241. doi:10.1007/s13174-011-0037-8

[5] M. Gavrilova, Ed., "Generalized Voronoi Diagram: A Geometry-Based Approach to Computational Intelligence (Studies in Computational Intelligence 158)," Springer, New York, 2008.

[6] W. A. Johnson and R. F. Mehl, "Reaction Kinetics in Processes of Nucleation and Growth," *Transactions of the American Institute of Mining, Metallurgical and Petroleum Engineers*, Vol. 135, 1939, pp. 416-458.

[7] J. Møller, "Lectures on Random Voronoi Tessellations. Lecture Notes in Statistics," Springer-Verlag, New York, 1994. doi:10.1007/978-1-4612-2652-9

[8] A. Okabe, B. Boots, K. Sugihara and S. N. Chiu, "Spatial Tessellations—Concepts and Applications of Voronoi Diagrams," 2nd Edition, John Wiley, Hoboken, 2000.

[9] K. Q. Brown, "Geometric Transforms for Fast Geometric Algorithms," Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, 1979.

[10] K. Q. Brown, "Voronoi Diagrams from Convex Hulls," *Information Processing Letters*, Vol. 9, No. 5, 1979, pp. 223-228. doi:10.1016/0020-0190(79)90074-7

[11] H.-S. Na, C.-N. Lee and O. Cheong, "Voronoi Diagrams on the Sphere," *Computational Geometry*: *Theory and Applications*, Vol. 23, No. 2, 2002, pp. 183-194. doi:10.1016/S0925-7721(02)00077-9